MB966

Intel® Xeon® / Core™ i3/i5/i7 Motherboard

USER'S MANUAL

Version 1.1

Acknowledgments

AMI is a registered trademark of American Megatrends, Inc.
PS/2 is a trademark of International Business Machines Corporation.
Intel is a registered trademark of Intel Corporation.
Microsoft Windows is a registered trademark of Microsoft Corporation.
Winbond is a registered trademark of Winbond Electronics Corporation.
All other product names or trademarks are properties of their respective owners.

Table of Contents

Introduction 1

Specifications 1
Checklist 3
Board Dimensions 4

Installations 6

Installing the CPU 7
Installing the Memory 8
Setting the Jumpers 9
Jumper Locations on MB966 10
Jumper Settings on MB966 11
JP1: TPM Enable/Disable Setting 11
JP2, JP3: Watchdog Timer & LAN Bypass Settings 11
JP4: Clear CMOS Contents 12
JP5: ME (Intel® Management Engine) Enable/Disable 12
JP7: PCIE1 & PCIE2 Golden Finger PCIe Configuration 12
Connectors on MB966 13
Connector Locations on MB966 14
USB_LAN1: 10/100/1000 RJ-45 (MB966) and USB1/2 Ports 15
ATX1: 24-pin ATX Power Connector 15
ATX_12V1: ATX 12V Power Connector 15
DIMM1, DIMM2: Channel A DDR3 Socket 15
DIMM3, DIMM4: Channel B DDR3 Socket 15
FAN1/2: CPU Fan Power Connectors 15
FAN3/4: System Fan Power Connectors 16
CN1: Front Side Bridge Board to ID370 16
CN2: Slim Type II Compact Flash Connector 16
CN3: COM1 Serial Port 17
J8: COM2 Serial Port 17
CN4: USB Connector 17
J3: USB Header 17
CN5: SGPIO Connector 17
CN6, CN9, CN11, CN12, CN13, CN14, CN15, CN16: LAN RJ-45 18
CN7, CN8, CN10: SATA HDD Connector 18
J1: LPC Debug Port (Factory use only) 18
J4: PS2 Keyboard/Mouse Connector 18
J5: SPI Debug Port (Factory use only) 18

J6: Mini-Card Connector 18
J7: VGA Box Header 19
J14: Power Output Connector 19
J15: Power Output Connector 19
LED GPIO Definition 20

Digital I/O Sample Configuration 21

Watchdog Timer Configuration 25

BIOS Setup 28

Drivers Installation 52

This page is intentionally left blank.

Introduction

Specifications

Product Name	MB966
Processor	Support for Intel® Lynnfield and Clarkdale processors which fall within mainstream TDP envelope, up to 95W TDP. Support for the following Intel® technologies: • Turbo Boost • Intel® 64 (64-bit extensions) • XD (Execution Disable Bit) • Hyper-Threading • VT-d (Intel® Virtualization Technology for Directed I/O) • TXT (Trusted Execution Technology)
Processor Speed	• Intel® Xeon X3450, X3430 • Intel® Core i7-860 • Intel® Core i5-750 • Intel® Core i5-660 (integrated graphics support) • Intel® Core i3-540 (integrated graphics support) • Intel® Pentium® Processor G6950 (integrated graphics support)
CPU VRD	Appropriate VRD for Lynnfield and Clarkdale
Chipset	Intel® 3450 PCH chip (single chip w/o ICH)
BIOS	AMI BIOS
Memory	• Dual channel DDR3 up to 1333MHz • Unbuffered with ECC or Unbuffered & non-ECC • Four DIMM sockets for max. 16GB memory
Video	• Intel® Core™ i5 / Pentium integrated
Network Controller	• MGN1: Intel® Nineveh Gigabit PHY (82578DM) w/ AMT support • LAN1~6: Intel® 82574L Gigabit Ethernet NIC Supports SNMP, PXE & Wake-on-LAN
Network Bypass	• Supports two bypass segments • Control by GPIO / Watchdog
SATA II	Intel® 3450 built-in SATA II controller (3.0Gb/sec), 3 ports
Compact Flash	CF Flash type II

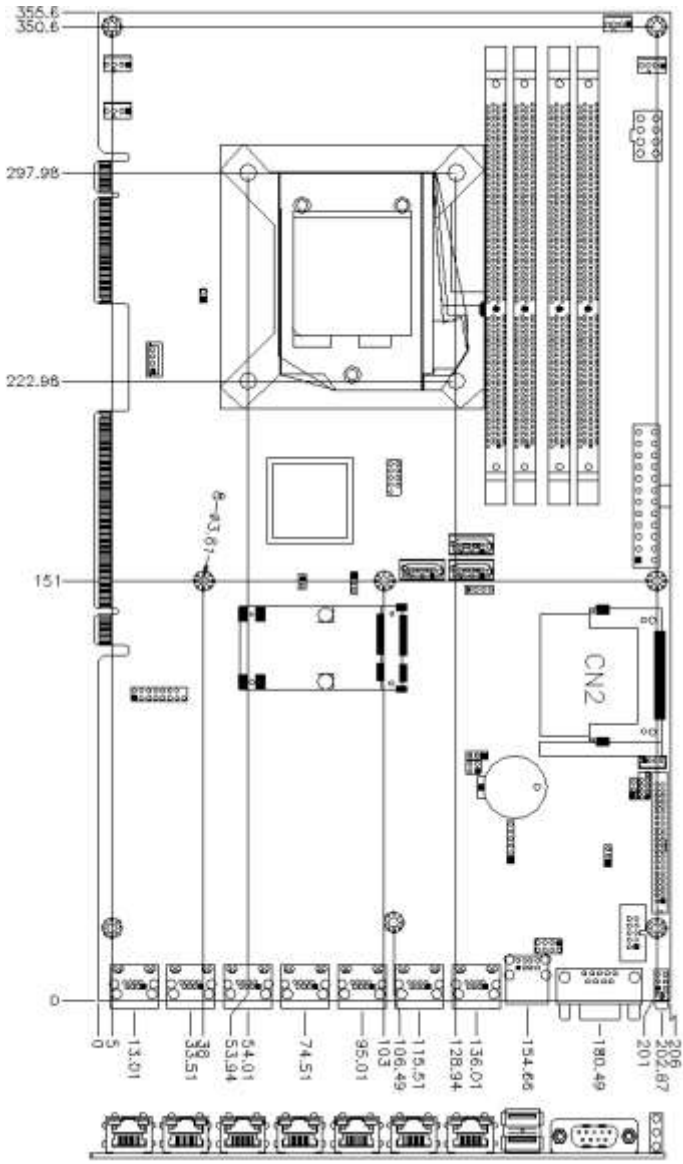
USB	Intel® 3450 built-in High Speed USB host controller, use 4 ports: • 2-port at front panel • 2-port USB by box header 2x4
LPC I/O	Winbond W83627DHG: • COM1 by pin header 1x8 • KB/Mouse by pin header 1x6 • Hardware monitor (3 thermal inputs, 4 voltage monitor inputs & 2 fan headers)
RTC	Intel® 3450 built-in RTC with on-board lithium battery & holder
Expansion	Two PCI-e x8 (Golden Finger) One Type II CF (Internal socket)
Front Panel LEDs	Status LED function as follows: #1 LED: System Power (Green, defined by S/W) #2 LED: Status2 / Alarm (Red / Green, defined by S/W) #3 LED: Status1 (Red / Yellow, defined by S/W)
Front Panel Connectors	• One Dual layer USB type A connector • One RJ-45 for Management port • Six RJ-45 with Link/Act, Speed LED for 10/100/1000M Ethernet • Three status LED
Fan Connectors	• 4 pins pin-header x2 for CPU fan (near CPU w/ tachometer sense) • 4 pins pin-header x2 for System fan (near rear side)
Power Connector	Supports 24pins + 8pins
Power Management	ACPI compliant – 5 levels: Full-on (S0) / Stop Grant (S1) / Suspend to Disk (S4) / Soft-off (S5)
Watchdog Timer	Yes (256 segments, 0, 1, 2…255 sec/min)
TPM	Winbond WPCT200 TPM1.2 controller x1 for Trust Platform 1.2
RoHS Compliant	Yes
MTBF	100,000
Board Size	8.11” x 14” (206 x 355.6 mm)

Checklist

Your MB966 package should include the items listed below.

- The MB966 motherboard
- This User's Manual

Board Dimensions



This page is intentionally left blank.

Installations

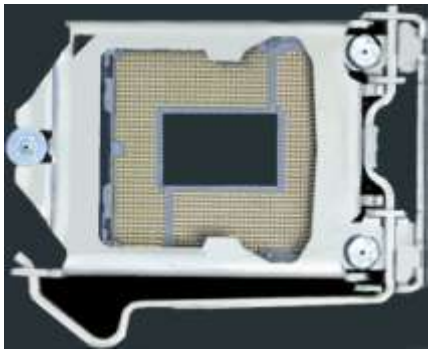
This section provides information on how to use the jumpers and connectors on the MB966 in order to set up a workable system. The topics covered are:

Installing the CPU 7
Installing the Memory 8
Setting the Jumpers 9
Connectors on MB966 13

Installing the CPU

The MB966 board supports an LGA1156 Socket (shown below) for Intel Clarkdale processors.

To install the CPU, unlock first the socket by pressing the lever sideways, then lift it up to a 90-degree. Then, position the CPU above the socket such that the CPU corner aligns with the gold triangle matching the socket corner with a small triangle. Carefully insert the CPU into the socket and push down the lever to secure the CPU. Then, install the heat sink and fan.



***NOTE**: Ensure that the CPU heat sink and the CPU top surface are in total contact to avoid CPU overheating problem that would cause your system to hang or be unstable.*

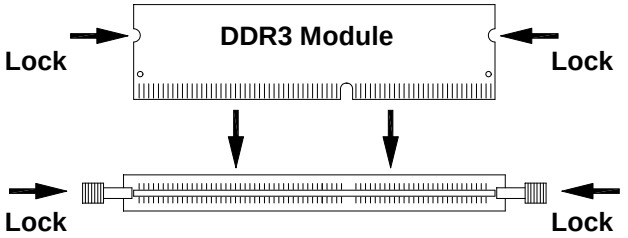
Installing the Memory

The MB966 board supports four DDR3 memory socket for a maximum total memory of 16GB in DDR3 DIMM memory type.

Installing and Removing Memory Modules

To install the DDR3 modules, locate the memory slot on the board and perform the following steps:

1. Hold the DDR3 module so that the key of the DDR3 module aligned with that on the memory slot.
2. Gently push the DDR3 module in an upright position until the clips of the slot close to hold the DDR3 module in place when the DDR3 module touches the bottom of the slot.
3. To remove the DDR3 module, press the clips with both hands.

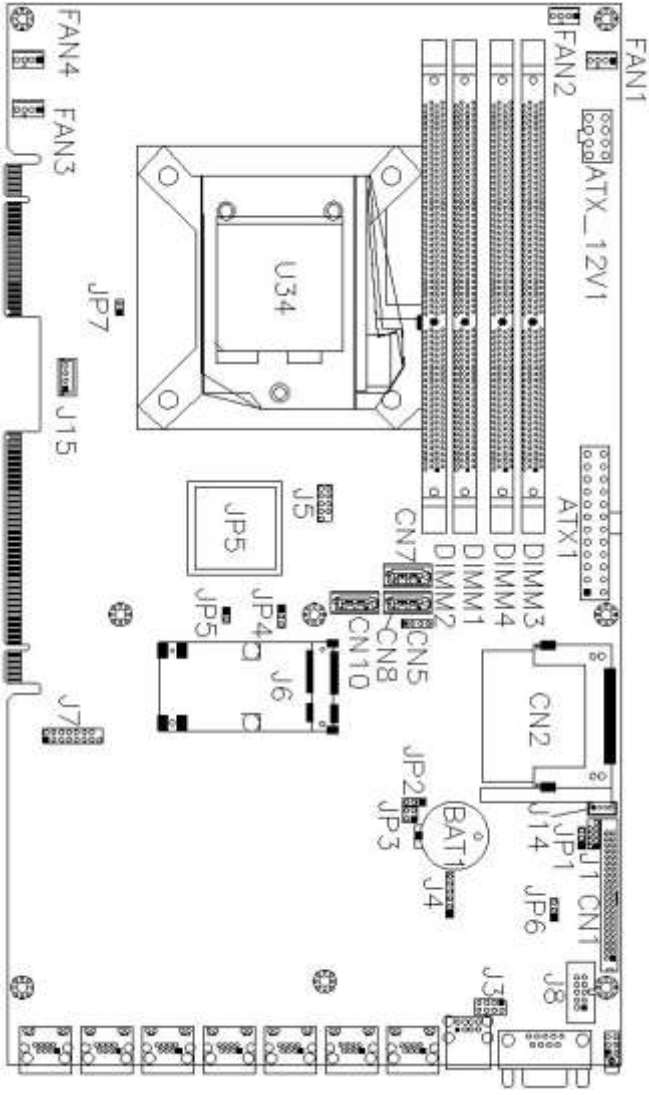


Setting the Jumpers

Jumpers are used on MB966 to select various settings and features according to your needs and applications. Contact your supplier if you have doubts about the best configuration for your needs. The following lists the connectors on MB966 and their respective functions.

Jumper Locations on MB966..... 10
Jumper Settings on MB966..... 11
JP1: TPM Enable/Disable Setting 11
JP2, JP3: Watchdog Timer & CN11~CN14 LAN Bypass Settings 11
JP4: Clear CMOS Contents 12
JP5: ME (Intel® Management Engine) Enable/Disable..... 12
JP7: PCIE1 & PCIE2 Golden Finger PCIe Configuration..... 12

Jumper Locations on MB966



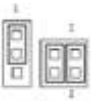
JP1: TPM Enable/Disable Setting
JP2, JP3: Watchdog Timer & CN11~CN14 LAN Bypass Settings
JP4: Clear CMOS Contents
JP5: ME (Intel® Management Engine) Enable/Disable
JP7: PCIE1 & PCIE2 Golden Finger PCIe Configuration

Jumper Settings on MB966

JP1: TPM Enable/Disable Setting

JP1	Setting	Function
	Pin 1-2 Short/Closed	Enable
	Pin 2-3 Short/Closed	Disable

JP2, JP3: Watchdog Timer & LAN Bypass Settings

JP2, JP3	Setting	Function	Power Off	Power On OS Run Software
	JP2 Pin 2-3 Closed JP3 Pin 1-2 Open & 3-4 Closed	System LAN bypass function is controlled by Super I/O GP23	LAN Bypass	GP23 Active: Low: Bypass High: Normal
		System will reboot upon the time out of watchdog timer.		WDT Reboot System
	JP2 Pin 1-2 Closed JP3 Pin 1-2 & 3-4 Open	System will bypass LAN upon the time out of watchdog timer.	LAN Bypass	
(Default)	JP2 Pin 2-3 Closed JP3 Pin 1-2 & 3-4 Open	System LAN bypass function is controlled by Super I/O GP23.		GP23 Active: Low: Bypass High: Normal
	JP2 Pin 1-2 Closed JP3 Pin 1-2 & 3-4 Closed	System LAN is at normal	LAN Bypass	LAN Always Normal
		System will reboot upon the time out of watchdog timer.		WDT Reboot System

JP4: Clear CMOS Contents

Use JP4 to clear the CMOS contents. *Note that the ATX-power connector should be disconnected from the board before clearing CMOS.*

JP4	Setting	Function
	Pin 1-2 Short/Closed	Normal
	Pin 2-3 Short/Closed	Clear CMOS

JP5: ME (Intel® Management Engine) Enable/Disable

JP5	Setting	Function
	Open	Enable
	Short/Closed	Disable

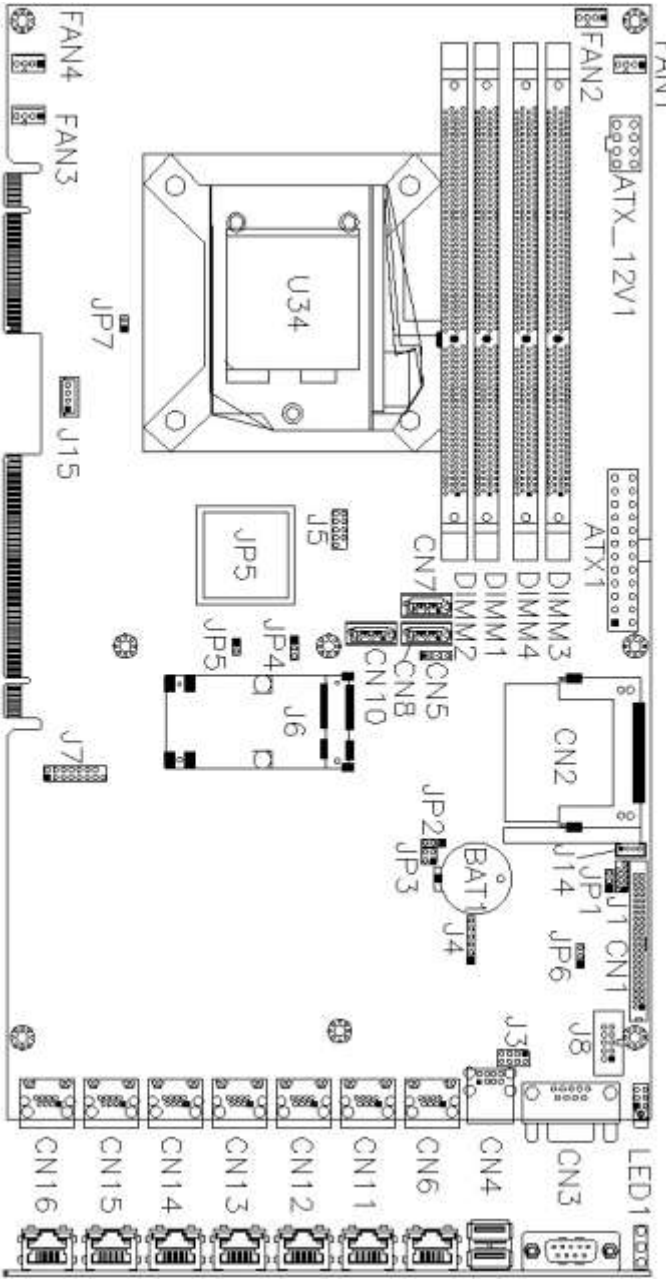
JP7: PCIE1 & PCIE2 Golden Finger PCIe Configuration

JP7	Setting	Function	Remarks
	Open	Combine to 1x16	For CPU with Integrated Graphics support
	Short/Closed	Separate to2x8	Default for CPU without Integrated Graphics support

Connectors on MB966

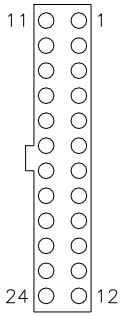
Connector Locations on MB966 14
Connector Locations on MB966 14
USB_LAN1: 10/100/1000 RJ-45 (MB966) and USB1/2 Ports 15
ATX1: 24-pin ATX Power Connector 15
ATX_12V1: ATX 12V Power Connector 15
DIMM1, DIMM2: Channel A DDR3 Socket 15
DIMM3, DIMM4: Channel B DDR3 Socket 15
FAN1/2/3/4: System Fan Power Connectors 15
CN1: Front Side Bridge Board to ID370 16
CN2: Slim Type II Compact Flash Connector 16
CN3: COM1 Serial Port 17
J8: COM2 Serial Port 17
CN4: USB Connector 17
J3: USB Header 17
CN5: SGPIO Connector 17
CN6, CN9, CN11, CN12, CN13, CN14, CN15, CN16: LAN RJ-45 18
CN7, CN8, CN9, CN10: SATA HDD Connector 18
J1: LPC Debug Port (Factory use only) 18
J4: PS2 Keyboard/Mouse Connector 18
J5: SPI Debug Port (Factory use only) 18
J6: Mini-Card Connector 18
J7: VGA Box Header 19
J14~J15: Power Output Connector 19

Connector Locations on MB966



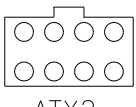
USB_LAN1: 10/100/1000 RJ-45 (MB966) and USB1/2 Ports

ATX1: 24-pin ATX Power Connector



Signal Name	Pin #	Pin #	Signal Name
3.3V	13	1	3.3V
-12V	14	2	3.3V
Ground	15	3	Ground
PS-ON	16	4	+5V
Ground	17	5	Ground
Ground	18	6	+5V
Ground	19	7	Ground
-5V	20	8	Power good
+5V	21	9	5VSB
+5V	22	10	+12V
+5V	23	11	+12V
Ground	24	12	+3.3V

ATX_12V1: ATX 12V Power Connector



Signal Name	Pin #	Pin #	Signal Name
+12V	5	1	Ground
+12V	6	2	Ground
+12V	7	3	Ground
+12V	8	4	Ground

DIMM1, DIMM2: Channel A DDR3 Socket

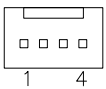
DIMM1 and DIMM2 are the first-channel DDR3 sockets.

DIMM3, DIMM4: Channel B DDR3 Socket

DIMM3 and DIMM4 are the second-channel DDR3 sockets.

FAN1/2: CPU Fan Power Connectors

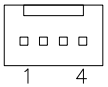
FAN1/2 are 4-pin headers for system fan power. The fan must be a 12V fan.



Pin #	Signal Name
1	Ground
2	+12V
3	Rotation detection
4	Control

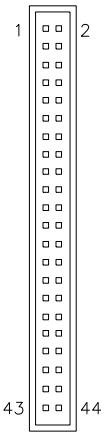
FAN3/4: System Fan Power Connectors

FAN3/4 are 4-pin headers for system fan power. The fan must be a 12V fan.



Pin #	Signal Name
1	Ground
2	+12V
3	Rotation detection
4	Control

CN1: Front Side Bridge Board to ID370

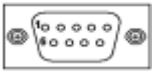


Signal Name	Pin #	Pin #	Signal Name
PWR-SW	1	2	GND
Reset-SW	3	4	GND
Speaker	5	6	VCC
VCC	7	8	VCC
GND (Power)	9	10	Pull up to VCC
HDD LED	11	12	Pull up to VCC
RELAY LED	13	14	Pull up to VCC
GND	15	16	GND
GND	17	18	GND
GPO0	19	20	GPI0
GPO1	21	22	GPI1
GPO2	23	24	GPI2
GPO3	25	26	GPI3
GPO4	27	28	GPI4
GPO5	29	30	GPI5
GPO6	31	32	GPI6
GPO7	33	34	GPI7
VCC	35	36	VCC
GPIO31	37	38	GPIO56
GPIO46	39	40	GPIO57
GPIO47	41	42	GPIO73
VCC3	43	44	VCC3

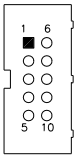
CN2: Slim Type II Compact Flash Connector

CN3: COM1 Serial Port

CN3 (COM1) is a DB-9 connector.



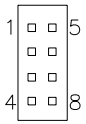
J8: COM2 Serial Port



Pin #	Signal Name (RS-232)
1	DCD, Data carrier detect
2	RXD, Receive data
3	TXD, Transmit data
4	DTR, Data terminal ready
5	Ground
6	DSR, Data set ready
7	RTS, Request to send
8	CTS, Clear to send
9	RI, Ring indicator
10	No Connect.

CN4: USB Connector

J3: USB Header



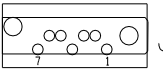
Signal Name	Pin #	Pin #	Signal Name
VCC	1	5	Ground
USB2-	2	6	USB3+
USB2+	3	7	USB3-
Ground	4	8	VCC

CN5: SGPIO Connector

Pin #	Signal Name
1	SCLOCK
2	SLOAD
3	SDATAOUT0
4	SDATAOUT1

CN6, CN9, CN11, CN12, CN13, CN14, CN15, CN16: LAN RJ-45

CN7, CN8, CN10: SATA HDD Connector



Pin #	Signal Name
1	Ground
2	TX+
3	TX-
4	Ground
5	RX-
6	RX+
7	Ground

J1: LPC Debug Port (Factory use only)

J4: PS2 Keyboard/Mouse Connector

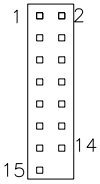
Pin #	Signal Name
1	KBDATA
2	KBCLK
3	MSDATA
4	MSCLK
5	GND
6	+5V

J5: SPI Debug Port (Factory use only)

J6: Mini-Card Connector

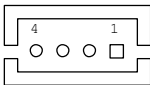
J7: VGA Box Header

(Available for Integrated Graphics in selected CPU only)



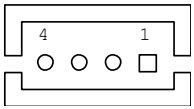
Signal Name	Pin	Pin	Signal Name
R	1	2	+5V
G	3	4	GND
B	5	6	NC
NC	7	8	SPD1
GND	9	10	Hsync
GND	11	12	Vsync
GND	13	14	SPCLK
GND	15		

J14: Power Output Connector



Pin #	Signal Name
1	+5V
2	GND
3	GND
4	+12V

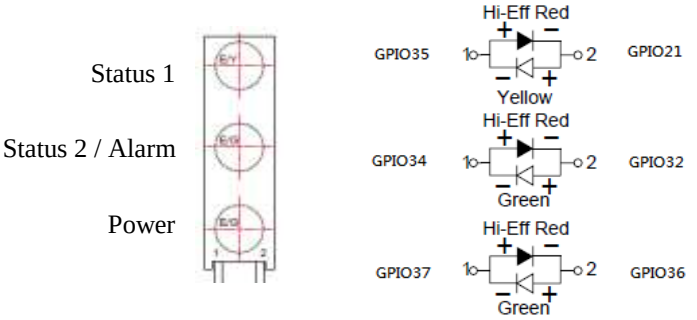
J15: Power Output Connector



Pin #	Signal Name
1	+5V
2	GND
3	GND
4	+12V

LED GPIO Definition

This chapter describes GPIO definition of three LEDs on front panel.



Status1 LED	GPIO35	GPIO21
RED	H	L
YELLOW	L	H

Status2 / Alarm LED	GPIO34	GPIO32
RED	H	L
GREEN	L	H

Power LED	GPIO37	GPIO36
RED	H	L
GREEN	L	H

Digital I/O Sample Configuration

Filename：Main.cpp

```
//---------------------------------------------------------------------------
//
// THIS CODE AND INFORMATION IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY
// KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE
// IMPLIED WARRANTIES OF MERCHANTABILITY AND/OR FITNESS FOR A PARTICULAR
// PURPOSE.
//
//---------------------------------------------------------------------------
#include <dos.h>
#include <conio.h>
#include <stdio.h>
#include <stdlib.h>
#include "W627DHG.H"
//---------------------------------------------------------------------------
int main (void);

void Dio2Initial(void);
void Dio2SetOutput(unsigned char);
unsigned char Dio2GetInput(void);
void Dio2SetDirection(unsigned char);
unsigned char Dio2GetDirection(void);

void Dio3Initial(void);
void Dio3SetOutput(unsigned char);
unsigned char Dio3GetInput(void);
void Dio3SetDirection(unsigned char);
unsigned char Dio3GetDirection(void);
//---------------------------------------------------------------------------
int main (void)
{
        char SIO;

        SIO = Init_W627DHG();
        if (SIO == 0)
        {
                printf("Can not detect Winbond 83627DHG, program abort.\n");
                return(1);
        }

        Dio2Initial();
        Dio3Initial();

        //for GPIO30..37
        Dio3SetDirection(0x0F);       //GP30..33 = input, GP34..37=output
        printf("Current DIO direction = 0x%X\n", Dio3GetDirection());

        printf("Current DIO status = 0x%X\n", Dio3GetInput());

        printf("Set DIO output to high\n");
        Dio3SetOutput(0x0F);

        printf("Set DIO output to low\n");
        Dio3SetOutput(0x00);

        return 0;
}
//---------------------------------------------------------------------------
void Dio2Initial(void)
{
        unsigned char ucBuf;

        //switch GPIO multi-function pin
        ucBuf = Get_W627DHG_Reg(0x24);
        ucBuf &= 0xFE;
        Set_W627DHG_Reg(0x24, ucBuf);

        Set_W627DHG_LD(0x09);                                   //switch to logic device 9

        //enable the GP2 group
```

```
	ucBuf = Get_W627DHG_Reg(0x30);
	ucBuf |= 0x01;
	Set_W627DHG_Reg(0x30, ucBuf);
}
//-----------------------------------------------------------------------------
void Dio2SetOutput(unsigned char)
{
	Set_W627DHG_LD(0x09);                          //switch to logic device 9
	Set_W627DHG_Reg(0xE4, NewData);
}
//-----------------------------------------------------------------------------
unsigned char Dio2GetInput(void)
{
	unsigned char result;

	Set_W627DHG_LD(0x09);                          //switch to logic device 9
	result = Get_W627DHG_Reg(0xE4);
	return (result);
}
//-----------------------------------------------------------------------------
void Dio2SetDirection(unsigned char)
{
	//NewData : 1 for input, 0 for output
	Set_W627DHG_LD(0x09);                          //switch to logic device 9
	Set_W627DHG_Reg(0xE3, NewData);
}
//-----------------------------------------------------------------------------
unsigned char Dio2GetDirection(void)
{
	unsigned char result;

	Set_W627DHG_LD(0x09);                          //switch to logic device 9
	result = Get_W627DHG_Reg(0xE0);
	return (result);
}
//-----------------------------------------------------------------------------
void Dio3Initial(void)
{
	unsigned char ucBuf;

	//switch GPIO multi-function pin
	ucBuf = Get_W627DHG_Reg(0x2C);
	ucBuf &= 0x1F;
	Set_W627DHG_Reg(0x2C, ucBuf);                  //clear

	Set_W627DHG_LD(0x09);                          //switch to logic device 9

	//enable the GP3 group
	ucBuf = Get_W627DHG_Reg(0x30);
	ucBuf |= 0x02;
	Set_W627DHG_Reg(0x30, ucBuf);

	//input detect type
	Set_W627DHG_Reg(0xFE, 0xFF);
}
//-----------------------------------------------------------------------------
void Dio3SetOutput(unsigned char NewData)
{
	Set_W627DHG_LD(0x09);                          //switch to logic device 9
	Set_W627DHG_Reg(0xF1, NewData);
}
//-----------------------------------------------------------------------------
unsigned char Dio3GetInput(void)
{
	unsigned char result;

	Set_W627DHG_LD(0x09);                          //switch to logic device 9
	result = Get_W627DHG_Reg(0xF1);
	return (result);
}
//-----------------------------------------------------------------------------
void Dio3SetDirection(unsigned char NewData)
{
```

```
        //NewData : 1 for input, 0 for output
        Set_W627DHG_LD(0x09);                                   //switch to logic device 9
        Set_W627DHG_Reg(0xF0, NewData);
}
//-----------------------------------------------------------------------------
unsigned char Dio3GetDirection(void)
{
        unsigned char result;

        Set_W627DHG_LD(0x09);                                   //switch to logic device 9
        result = Get_W627DHG_Reg(0xF0);
        return (result);
}
//-----------------------------------------------------------------------------
```

Filename：W627DHG.cpp

```
//-----------------------------------------------------------------------------
//
// THIS CODE AND INFORMATION IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY
// KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE
// IMPLIED WARRANTIES OF MERCHANTABILITY AND/OR FITNESS FOR A PARTICULAR
// PURPOSE.
//
//-----------------------------------------------------------------------------
#include "W627DHG.H"
#include <dos.h>
//-----------------------------------------------------------------------------
unsigned int W627DHG_BASE;
void Unlock_W627DHG (void);
void Lock_W627DHG (void);
//-----------------------------------------------------------------------------
unsigned int Init_W627DHG(void)
{
        unsigned int result;
        unsigned char ucDid;

        W627DHG_BASE = 0x4E;
        result = W627DHG_BASE;

        ucDid = Get_W627DHG_Reg(0x20);
        if (ucDid == 0xA0)                                      //W83627DHG
        {       goto Init_Finish;       }
        else if (ucDid == 0xB0)                                 //W83627DHG-P
        {       goto Init_Finish;       }

        W627DHG_BASE = 0x2E;
        result = W627DHG_BASE;

        ucDid = Get_W627DHG_Reg(0x20);
        if (ucDid == 0xA0)                                      //W83627DHG
        {       goto Init_Finish;       }
        else if (ucDid == 0xB0)                                 //W83627DHG-P
        {       goto Init_Finish;       }

        W627DHG_BASE = 0x00;
        result = W627DHG_BASE;

Init_Finish:
        return (result);
}
//-----------------------------------------------------------------------------
void Unlock_W627DHG (void)
{
        outportb(W627DHG_INDEX_PORT, W627DHG_UNLOCK);
        outportb(W627DHG_INDEX_PORT, W627DHG_UNLOCK);
}
//-----------------------------------------------------------------------------
void Lock_W627DHG (void)
{
        outportb(W627DHG_INDEX_PORT, W627DHG_LOCK);
}
//-----------------------------------------------------------------------------
```

```
void Set_W627DHG_LD( unsigned char LD)
{
	Unlock_W627DHG();
	outportb(W627DHG_INDEX_PORT, W627DHG_REG_LD);
	outportb(W627DHG_DATA_PORT, LD);
	Lock_W627DHG();
}
//----------------------------------------------------------------------------
void Set_W627DHG_Reg( unsigned char REG, unsigned char DATA)
{
	Unlock_W627DHG();
	outportb(W627DHG_INDEX_PORT, REG);
	outportb(W627DHG_DATA_PORT, DATA);
	Lock_W627DHG();
}
//----------------------------------------------------------------------------
unsigned char Get_W627DHG_Reg(unsigned char REG)
{
	unsigned char Result;
	Unlock_W627DHG();
	outportb(W627DHG_INDEX_PORT, REG);
	Result = inportb(W627DHG_DATA_PORT);
	Lock_W627DHG();
	return Result;
}
//----------------------------------------------------------------------------
```

Filename：W627DHG.h

```
//----------------------------------------------------------------------------
//
// THIS CODE AND INFORMATION IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY
// KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE
// IMPLIED WARRANTIES OF MERCHANTABILITY AND/OR FITNESS FOR A PARTICULAR
// PURPOSE.
//
//----------------------------------------------------------------------------
#ifndef __W627DHG_H
#define __W627DHG_H			1
//----------------------------------------------------------------------------
#define W627DHG_INDEX_PORT		(W627DHG_BASE)
#define W627DHG_DATA_PORT		(W627DHG_BASE+1)
//----------------------------------------------------------------------------
#define W627DHG_REG_LD			0x07
//----------------------------------------------------------------------------
#define W627DHG_UNLOCK			0x87
#define W627DHG_LOCK			0xAA
//----------------------------------------------------------------------------
unsigned int Init_W627DHG(void);
void Set_W627DHG_LD( unsigned char);
void Set_W627DHG_Reg( unsigned char, unsigned char);
unsigned char Get_W627DHG_Reg( unsigned char);
//----------------------------------------------------------------------------
#endif  //__W627DHG_H
```

Watchdog Timer Configuration

The WDT is used to generate a variety of output signals after a user programmable count. The WDT is suitable for use in the prevention of system lock-up, such as when software becomes trapped in a deadlock. Under these sorts of circumstances, the timer will count to zero and the selected outputs will be driven. Under normal circumstance, the user will restart the WDT at regular intervals before the timer counts to zero.

SAMPLE CODE:

This code and information is provided "as is" without warranty of any kind, either expressed or implied, including but not limited to the implied warranties of merchantability and/or fitness for a particular purpose.

Filename：Main.cpp

```
//---------------------------------------------------------------------------
//
// THIS CODE AND INFORMATION IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY
// KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE
// IMPLIED WARRANTIES OF MERCHANTABILITY AND/OR FITNESS FOR A PARTICULAR
// PURPOSE.
//
//---------------------------------------------------------------------------
#include <dos.h>
#include <conio.h>
#include <stdio.h>
#include <stdlib.h>
#include "W627DHG.H"
//---------------------------------------------------------------------------
int main (void);

void WDTInitial(void);
void WDTEnable(unsigned char);
void WDTDisable(void);

//---------------------------------------------------------------------------
int main (void)
{
	char SIO;

	SIO = Init_W627DHG();
	if (SIO == 0)
	{
		printf("Can not detect Winbond 83627DHG, program abort.\n");
		return(1);
	}

	WDTInitial();

	WDTEnable(10);

	WDTDisable();

	return 0;
}
//---------------------------------------------------------------------------
void WDTInitial(void)
{
	unsigned char bBuf;

	bBuf = Get_W627DHG_Reg(0x2D);
	bBuf &= (~0x01);
	Set_W627DHG_Reg(0x2D, bBuf);			//Enable WDTO
}
//---------------------------------------------------------------------------
```

```
void WDTEnable(unsigned char NewInterval)
{
        unsigned char bBuf;

        Set_W627DHG_LD(0x08);                                   //switch to logic device 8
        Set_W627DHG_Reg(0x30, 0x01);                            //enable timer

        bBuf = Get_W627DHG_Reg(0xF5);
        bBuf &= (~0x08);
        Set_W627DHG_Reg(0xF5, bBuf);                            //count mode is second

        Set_W627DHG_Reg(0xF6, NewInterval);                     //set timer
}
//-----------------------------------------------------------------------------
void WDTDisable(void)
{
        Set_W627DHG_LD(0x08);                                   //switch to logic device 8
        Set_W627DHG_Reg(0xF6, 0x00);                            //clear watchdog timer
        Set_W627DHG_Reg(0x30, 0x00);                            //watchdog disabled
}
//-----------------------------------------------------------------------------
```

Filename：W627DHG.cpp

```
//-----------------------------------------------------------------------------
//
// THIS CODE AND INFORMATION IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY
// KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE
// IMPLIED WARRANTIES OF MERCHANTABILITY AND/OR FITNESS FOR A PARTICULAR
// PURPOSE.
//
//-----------------------------------------------------------------------------
#include "W627DHG.H"
#include <dos.h>
//-----------------------------------------------------------------------------
unsigned int W627DHG_BASE;
void Unlock_W627DHG (void);
void Lock_W627DHG (void);
//-----------------------------------------------------------------------------
unsigned int Init_W627DHG(void)
{
        unsigned int result;
        unsigned char ucDid;

        W627DHG_BASE = 0x4E;
        result = W627DHG_BASE;

        ucDid = Get_W627DHG_Reg(0x20);
        if (ucDid == 0xA0)                                      //W83627DHG
        {       goto Init_Finish;       }
        else if (ucDid == 0xB0)                                 //W83627DHG-P
        {       goto Init_Finish;       }

        W627DHG_BASE = 0x2E;
        result = W627DHG_BASE;

        ucDid = Get_W627DHG_Reg(0x20);
        if (ucDid == 0xA0)                                      //W83627DHG
        {       goto Init_Finish;       }
        else if (ucDid == 0xB0)                                 //W83627DHG-P
        {       goto Init_Finish;       }

        W627DHG_BASE = 0x00;
        result = W627DHG_BASE;

Init_Finish:
        return (result);
}
//-----------------------------------------------------------------------------
void Unlock_W627DHG (void)
{
        outportb(W627DHG_INDEX_PORT, W627DHG_UNLOCK);
        outportb(W627DHG_INDEX_PORT, W627DHG_UNLOCK);
```

```
}
//---------------------------------------------------------------------------
void Lock_W627DHG (void)
{
        outportb(W627DHG_INDEX_PORT, W627DHG_LOCK);
}
//---------------------------------------------------------------------------
void Set_W627DHG_LD( unsigned char LD)
{
        Unlock_W627DHG();
        outportb(W627DHG_INDEX_PORT, W627DHG_REG_LD);
        outportb(W627DHG_DATA_PORT, LD);
        Lock_W627DHG();
}
//---------------------------------------------------------------------------
void Set_W627DHG_Reg( unsigned char REG, unsigned char DATA)
{
        Unlock_W627DHG();
        outportb(W627DHG_INDEX_PORT, REG);
        outportb(W627DHG_DATA_PORT, DATA);
        Lock_W627DHG();
}
//---------------------------------------------------------------------------
unsigned char Get_W627DHG_Reg(unsigned char REG)
{
        unsigned char Result;
        Unlock_W627DHG();
        outportb(W627DHG_INDEX_PORT, REG);
        Result = inportb(W627DHG_DATA_PORT);
        Lock_W627DHG();
        return Result;
}
//---------------------------------------------------------------------------
```

Filename：W627DHG.h

```
//---------------------------------------------------------------------------
//
// THIS CODE AND INFORMATION IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY
// KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE
// IMPLIED WARRANTIES OF MERCHANTABILITY AND/OR FITNESS FOR A PARTICULAR
// PURPOSE.
//
//---------------------------------------------------------------------------
#ifndef __W627DHG_H
#define __W627DHG_H                     1
//---------------------------------------------------------------------------
#define W627DHG_INDEX_PORT              (W627DHG_BASE)
#define W627DHG_DATA_PORT               (W627DHG_BASE+1)
//---------------------------------------------------------------------------
#define W627DHG_REG_LD                  0x07
//---------------------------------------------------------------------------
#define W627DHG_UNLOCK                  0x87
#define W627DHG_LOCK                    0xAA
//---------------------------------------------------------------------------
unsigned int Init_W627DHG(void);
void Set_W627DHG_LD( unsigned char);
void Set_W627DHG_Reg( unsigned char, unsigned char);
unsigned char Get_W627DHG_Reg( unsigned char);
//---------------------------------------------------------------------------
#endif  //__W627DHG_H
```

BIOS Setup

This chapter describes the different settings available in the AMI BIOS that comes with the board. The topics covered in this chapter are as follows:

BIOS Introduction 29
BIOS Setup 29
Main BIOS Setup 30
Advanced Settings 31
PCIPnP Settings 43
Boot Settings 45
Security Settings 46
Advanced Chipset Settings 47
Exit Setup 50
Load Optimal Defaults 50
Load Failsafe Defaults 50

BIOS Introduction

The BIOS (Basic Input/Output System) installed in your computer system's ROM supports Intel processors. The BIOS provides critical low-level support for a standard device such as disk drives, serial ports and parallel ports. It also adds virus and password protection as well as special support for detailed fine-tuning of the chipset controlling the entire system.

BIOS Setup

The BIOS provides a Setup utility program for specifying the system configurations and settings. The BIOS ROM of the system stores the Setup utility. When you turn on the computer, the BIOS is immediately activated. Pressing the <Del> key immediately allows you to enter the Setup utility. If you are a little bit late pressing the <Del> key, POST (Power On Self Test) will continue with its test routines, thus preventing you from invoking the Setup. If you still wish to enter Setup, restart the system by pressing the "Reset" button or simultaneously pressing the <Ctrl>, <Alt> and <Delete> keys. You can also restart by turning the system Off and back On again. The following message will appear on the screen:

```
Press  <DEL>  to  Enter  Setup
```

In general, you press the arrow keys to highlight items, <Enter> to select, the <PgUp> and <PgDn> keys to change entries, <F1> for help and <Esc> to quit.

When you enter the Setup utility, the Main Menu screen will appear on the screen. The Main Menu allows you to select from various setup functions and exit choices.

Main BIOS Setup

This setup allows you to view processor configuration used in your computer system and set the system time and date.

BIOS SETUP UTILITY

Main	Advanced	PCIPnP	Boot	Security	Chipset	Exit

Processor		**Use[ENTER], [TAB] or [SHIFT-TAB] to select a field.**
Intel(R) Core(TM) i5 CPU	660 @ 3.33GHz	
Speed : 3333MHz		**Use [+] or [-] to configure system Time.**
Count : 1		
System Memory		
Size : 8056MB		
System Time	**[02:29:50]**	<- Select Screen
System Date	[Fri 01/02/2009]	↑ ↓ Select Item
		+- Change Field
		Tab Select Field
		F1 General Help
		F10 Save and Exit
		ESC Exit

Note: *If the system cannot boot after making and saving system changes with Setup, the AMI BIOS supports an override to the CMOS settings that resets your system to its default.*

Warning: *It is strongly recommended that you avoid making any changes to the chipset defaults. These defaults have been carefully chosen by both AMI and your system manufacturer to provide the absolute maximum performance and reliability. Changing the defaults could cause the system to become unstable and crash in some cases.*

Advanced Settings

BIOS SETUP UTILITY

Main	**Advanced**	PCIPnP	Boot	Security	Chipset	Exit

Advanced Settings

WARNING: Setting wrong values in below sections may cause system to malfunction.

- ► **CPU Configurations**
- ► IDE Configuration
- ► SuperIO Configuration
- ► Hardware Health Configuration
- ► ACPI Configuration
- ► AHCI Configuration
- ► Event Log Configuration
- ► Intel AMT Configuration
- ► Intel VT-d Configuration
- ► MPS Configuration
- ► PCI Express Configuration
- ► Remote Access Configuration
- ► Trusted Computing
- ► USB Configuration
- ► Clock Generator Configuration

Configure CPU.

<- Select Screen
↑↓ Select Item
+- Change Field
Tab Select Field
F1 General Help
F10 Save and Exit
ESC Exit

The Advanced BIOS Settings configurations are shown in the following pages, as seen in the computer screen. Please note that setting the wrong values may cause the system to malfunction.

BIOS SETUP UTILITY

Main	**Advanced**	PCIPnP	Boot	Security	Chipset	Exit

Configure advanced CPU settings

Module Version: 01.08

Manufacturer: Intel	
Intel(R) Core(TM) i5 CPU	660 @ 3.33GHz
Frequency : 3.33GHz	
BLCK Speed : 133MHz	
Cache L1 : 128KB	
Cache L2 : 512KB	
Cache L3 : 4096KB	
Ratio Status: Unlocked (Min:09, Max:25)	
Ratio Actual Value: 9.5	
Ratio CMOS Setting	25
MPS and ACPI MADT ordering	Modern ordering
Max CPUID Value Limit	Disabled
Intel(R) Virtualization Tech	Disabled
Active Processor Cores	All
A20M	Diabled
►Intel PPM Configuration	

Configure CPU.

<- Select Screen
↑↓ Select Item
+- Change Field
Tab Select Field
F1 General Help
F10 Save and Exit
ESC Exit

The CPU Configuration menu shows the following CPU details including the manufacturer, CPU type, its frequency and cache levels. Other options include:

Ratio CMOS Setting
Sets the ratio between CPU core clock and the FSB frequency.

MPS and ACPI MADT ordering
Modern ordering for Windows XP or later OSes. Legacy ordering for Windows 2000 or earlier OSes.

Max CPU ID Value Limit
Disabled for Windows XP.

Intel Virtualization Tech
When enabled, a VMM can utilize the additional HW Caps. Provided by Intel Vitualization Tech. Note: A full reset is required to change the setting.

Active Processor Cores
Number of cores to enable in each processor package.

A20M
Legacy OSes and Aps may need A20 M enabled.

Intel PPM Configuration
This configuration includes the following options:

Intel SpeedStep tech
Disable: Disable GV3 Enable: Enable GV3

Intel TurboMode tech
Turbo mode allows processor cores to run faster than marked frequency in specific condition.

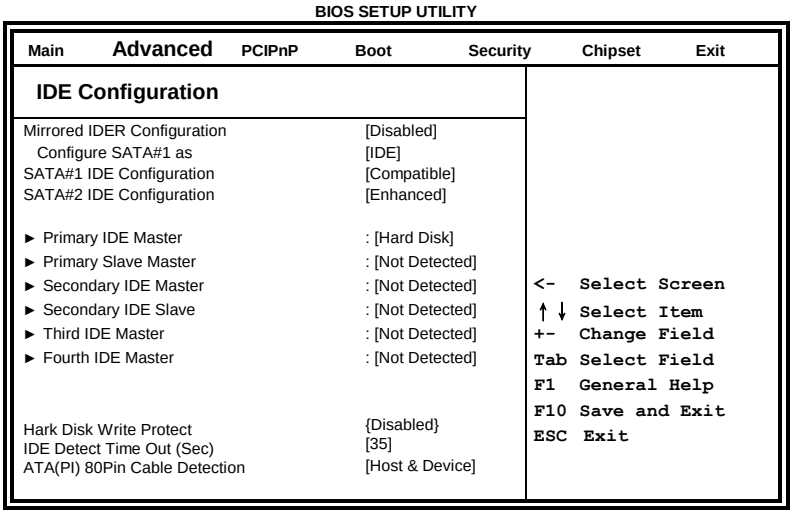
Intel C-STATE tech
CState: CPU idle is set to C2/C3/C4.

C State package limit setting
Selected option will program into C State package limit register.

C3 State / C6 State
Nehalem C state action select.

C1 Auto Demotion
When enabled, CPU will conditionally demote C3/C6/C7 requests to C1 based on uncore auto-demote information.

C3 Auto Demotion
When enabled, CPU will conditionally demote C6/C7 requests to C3 based on uncore auto-demote information.


BIOS SETUP UTILITY
Main Advanced PCIPnP Boot Security Chipset Exit
IDE Configuration
Mirrored IDER Configuration [Disabled]
Configure SATA#1 as [IDE]
SATA#1 IDE Configuration [Compatible]
SATA#2 IDE Configuration [Enhanced]
► Primary IDE Master : [Hard Disk]
► Primary Slave Master : [Not Detected]
► Secondary IDE Master : [Not Detected]
► Secondary IDE Slave : [Not Detected]
► Third IDE Master : [Not Detected]
► Fourth IDE Master : [Not Detected]
Hark Disk Write Protect {Disabled}
IDE Detect Time Out (Sec) [35]
ATA(PI) 80Pin Cable Detection [Host & Device]
<- Select Screen
↑ ↓ Select Item
+- Change Field
Tab Select Field
F1 General Help
F10 Save and Exit
ESC Exit

The IDE Configuration menu is used to change and/or set the configuration of the IDE devices installed in the system.

Hard Disk Write Protect

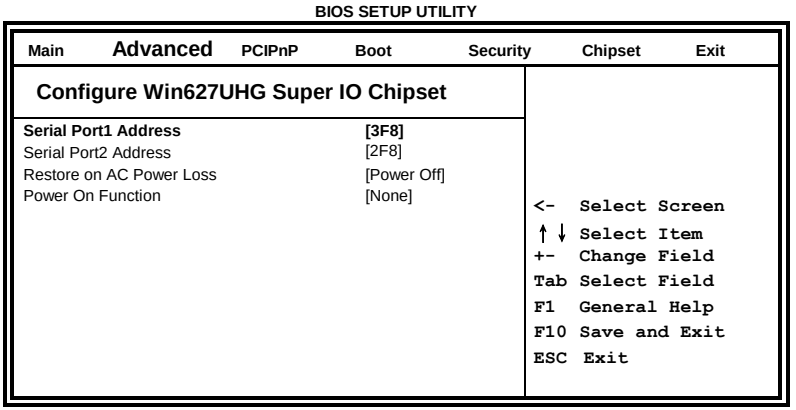
Disable/Enable device write protection. This will be effective only if device is accessed through BIOS.

IDE Detect Time Out (Sec)

Select the time out value for detecting ATA/ATAPI device(s).

ATA(PI) 80pin Cable Detection

Select the mechanism for detecting 80pin ATA(PI) cable.



Onboard Serial Port/Parallel Port

These fields allow you to select the onboard serial ports and their addresses. The default values for these ports are:

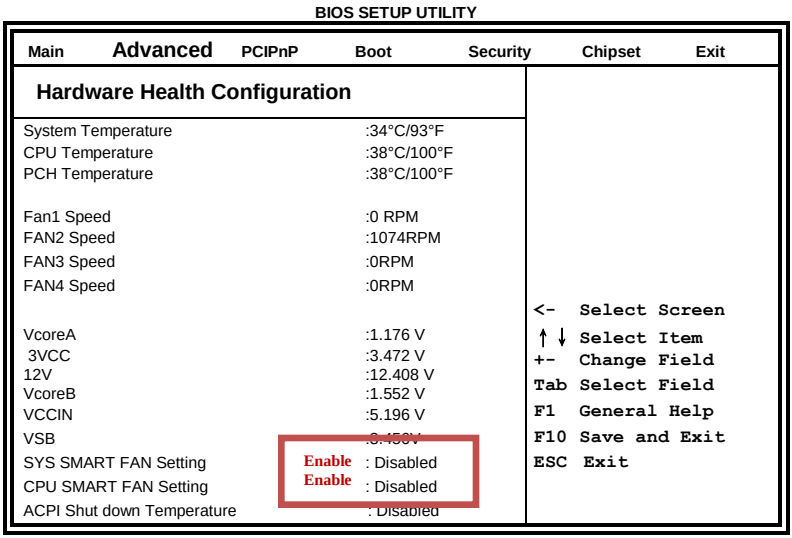
Serial Port 1 3F8
Serial Port 2 2F8/

Restore on AC Power Loss

This field sets the system power status whether *on or off* when power returns to the system from a power failure situation.

Power On Function

This field is related to how the system is powered on. The options are *None, Mouse Left, Mouse Right,* and *Any Key*.



The Hardware Health Configuration menu is used to show the operating temperature, fan speeds and system voltages.

SYS smart fan

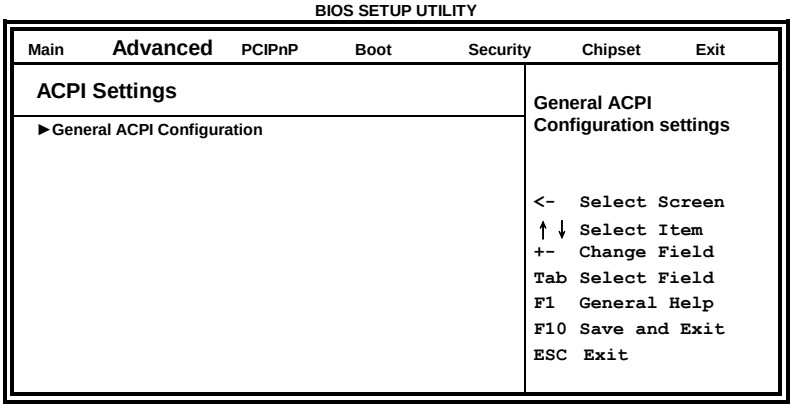
The options are *Disabled and Enabled (Default)*

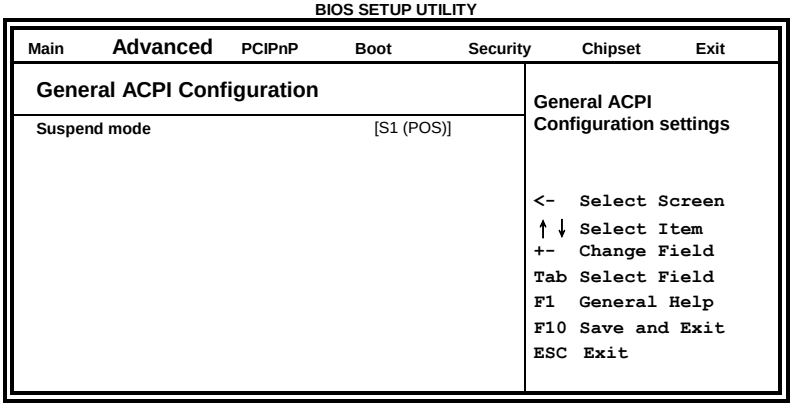
CPU smart fan

The options are *Disabled and Enabled (Default)*

ACPI Shutdown Temperature

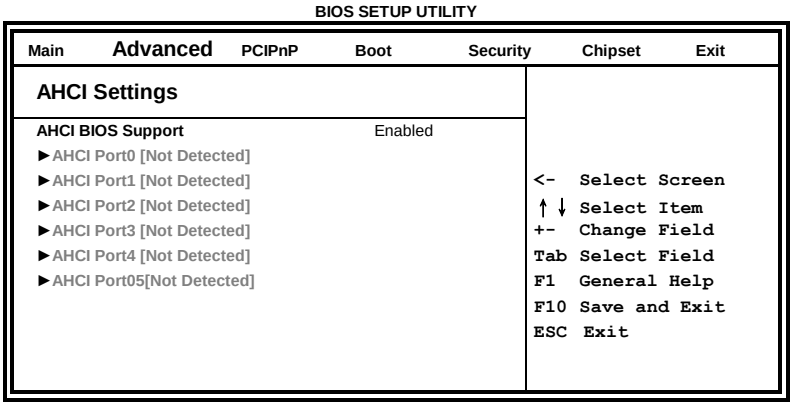
The options are *Disabled, 70℃/158℉, 75℃/167℉, 80℃/176℉, 85℃/185℉, 90℃/194℉,* and *95℃/203℉*.





Suspend Mode
The options of this field are *S1*, *S3*.

Repost Video on S3 Resume
Determines whether to invoke VGA BIOS post on S3/STR resume.

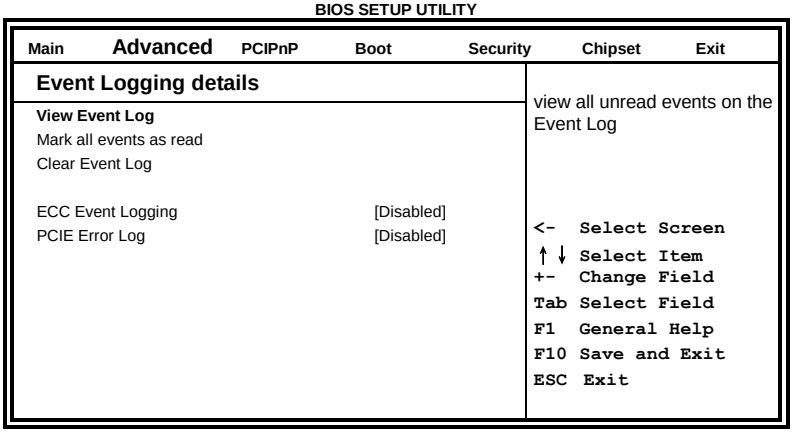


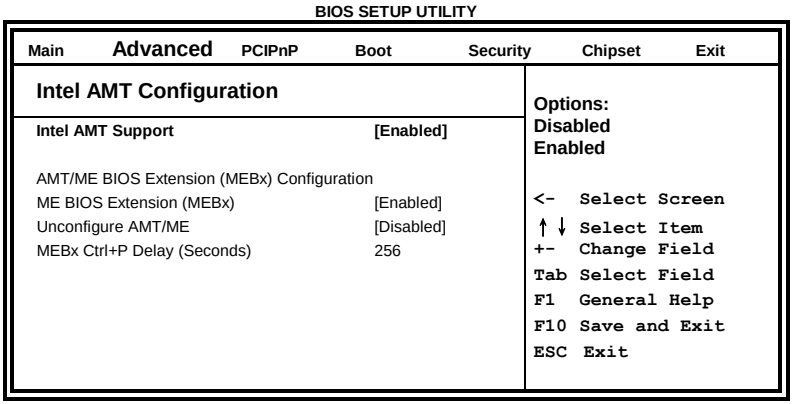
AHCI BIOS Support

Enables for supporting AHCI controller operates in AHCI mode during BIOS control otherwise operates in IDE mode

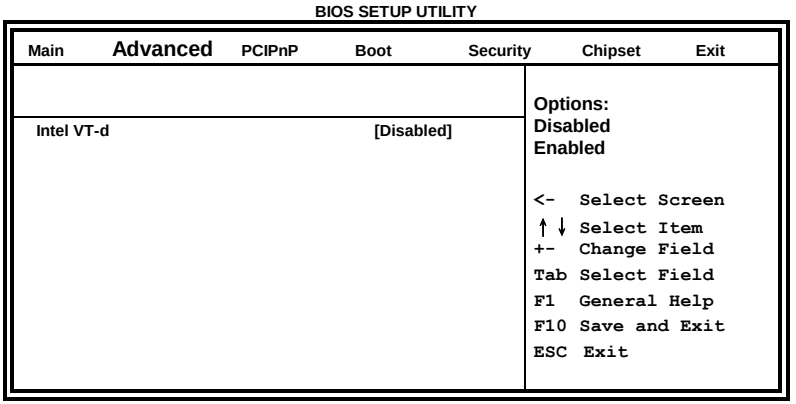
AHCI Port

While entering setup, BIOS auto detects the presence of IDE devices. This displays the status of auto detection of IDE devices.



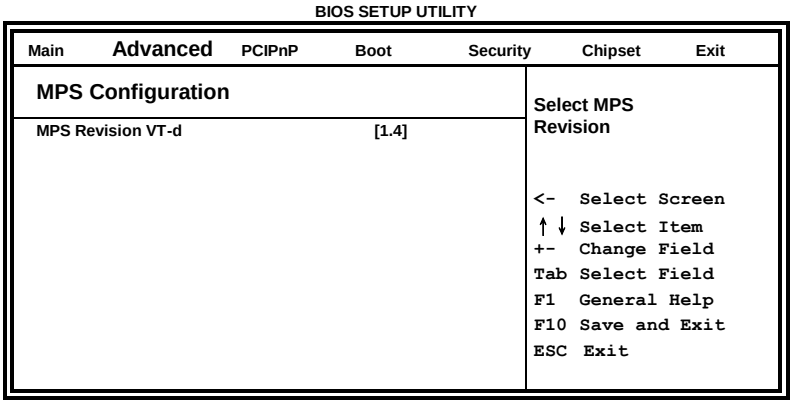


The Intel AMT Configuration configures the Intel Active Management Technology (AMT) options.



VT-d

Virtualization solutions allow multiple operating systems and applications to run in independent partitions all on a single computer. Using virtualization capabilities, one physical computer system can function as multiple "virtual" systems.



MPS Version Control for OS

This option is specifies the MPS (Multiprocessor Specification) version for your operating system. MPS version 1.4 added extended configuration tables to improve support for multiple PCI bus configurations and improve future expandability.
The default setting is ***1.4***.

BIOS SETUP UTILITY

Main	Advanced	PCIPnP	Boot	Security	Chipset	Exit

PCI Express Configuration

Active State Power Management	**[Disabled]**

Enable/Disable PCI Express L0s and L1 link power states

<- Select Screen
↑ ↓ Select Item
+- Change Field
Tab Select Field
F1 General Help
F10 Save and Exit
ESC Exit

BIOS SETUP UTILITY

Main	Advanced	PCIPnP	Boot	Security	Chipset	Exit

Configure Remote Access type and parameters

Remote Access	**Enabled**
Serial port number	**[COM1]**
Base Address, IRQ	**[3F8h, 4]**
Serial Port Mode	[1115200 8,n,1]
Flow Control	[None]
Redirection After BIOS POST	**Always**
Terminal Type	ANSI
VT-UTF8 Combo Key Support	Enabled
Sredir Memory Display Delay	No Delay

Select Remote Access type.

<- Select Screen
↑ ↓ Select Item
+- Change Field
Tab Select Field
F1 General Help
F10 Save and Exit
ESC Exit

When enabled, the Remote Acces type and parameters are shown:

Serial port number - Select Serial Port for console redirection.

Serial port mode - Select Serial Port settings.

Flow Control - Select Flow Control for console redirection.

Redirection After BIOS POST

Disable: Turns off the redirection after POST.

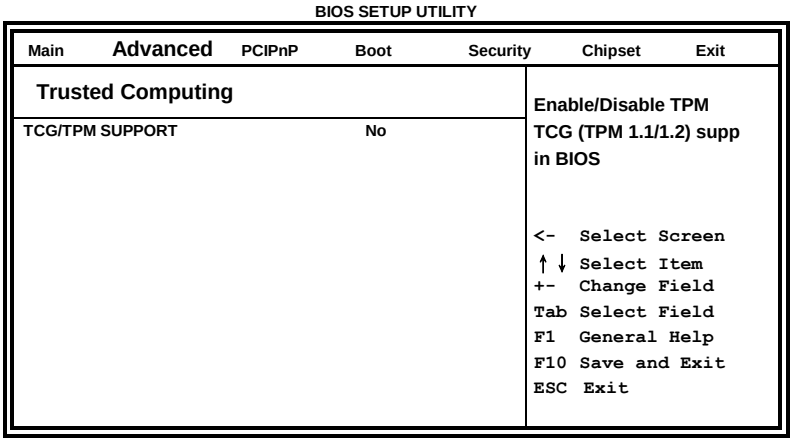
Boot Loader: Redirection is active during POST and during Boot Loader.

Always: Redirection is always active. (Some OSs may not work if set to Always.)

Terminal Type - Select the target terminal type.

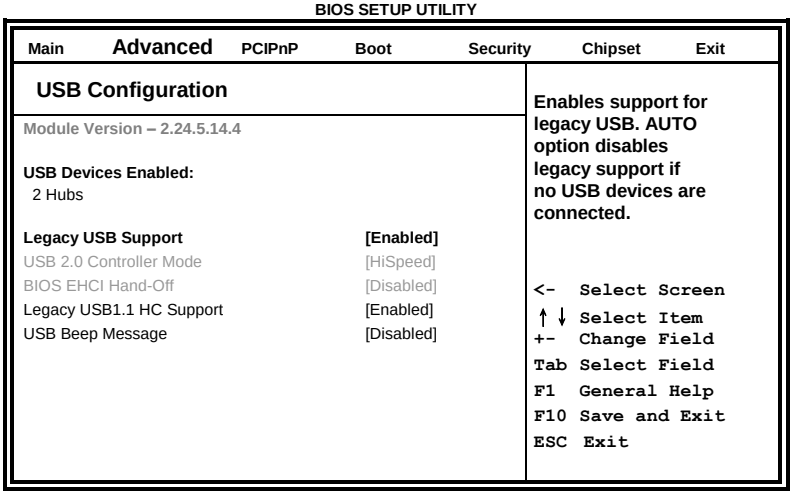
VT-UTF8 Combo Key Support – Enable VT-UTF8 Combination Key Support for ANSI/VT100 terminals.

Sredir Memory Display Delay – Gives the delay in seconds to display memory information.



USB Configuration

This option is used to configure USB mass storage class devices.


BIOS SETUP UTILITY
Main Advanced PCIPnP Boot Security Chipset Exit
USB Configuration
Module Version – 2.24.5.14.4
USB Devices Enabled:
2 Hubs
Legacy USB Support [Enabled]
USB 2.0 Controller Mode [HiSpeed]
BIOS EHCI Hand-Off [Disabled]
Legacy USB1.1 HC Support [Enabled]
USB Beep Message [Disabled]
Enables support for legacy USB. AUTO option disables legacy support if no USB devices are connected.
<- Select Screen
↑↓ Select Item
+- Change Field
Tab Select Field
F1 General Help
F10 Save and Exit
ESC Exit

Legacy USB Support

Enables support for legacy USB. AUTO option disables legacy support if no USB devices are connected.

Legacy USB1.1 HC Support

Support USB 1.1 HC.

USB Beep Message

Enables the beep during USB device enumeration.

Clock Generator Configuration

BIOS SETUP UTILITY

Main	Advanced	PCIPnP	Boot	Security	Chipset	Exit

Spectrum Enable/Disable	[Disable]	Epectrum Enable/Disable
		<- Select Screen
		↑ ↓ Select Item
		+- Change Field
		Tab Select Field
		F1 General Help
		F10 Save and Exit
		ESC Exit

PCIPnP Settings

BIOS SETUP UTILITY

Main	Advanced	PCIPnP	Boot	Security	Chipset	Exit

Advanced PCI/PnP Settings

WARNING: Setting wrong values in below sections may cause system to malfunction.

Clear NVRAM	**[No]**
Plug & Play O/S	[No]
PCI Latency Timer	[64]
Allocate IRQ to PCI VGA	[Yes]
Palette Snooping	[Disabled]
PCI IDE BusMaster	[Enabled]
OffBoard PCI/ISA IDE Card	**[Auto]**
IRQ3	[Available]
IRQ4	[Available]
IRQ5	[Available]
IRQ7	[Available]
IRQ9	[Available]
IRQ10	[Available]
IRQ11	[Available]
IRQ14	[Available]
IRQ15	[Available]
DMA Channel 0	[Available]
DMA Channel 1	[Available]
DMA Channel 3	[Available]
DMA Channel 5	[Available]
DMA Channel 6	[Available]
DMA Channel 7	[Available]
Reserved Memory Size	**[Disabled]**

Clear NVRAM during System Boot

<- Select Screen
↑ ↓ Select Item
+- Change Field
Tab Select Field
F1 General Help
F10 Save and Exit
ESC Exit

Clear NVRAM

This item is used for clearing NVRAM during system boot.

Plug & Play O/S

This lets BIOS configure all devices in the system or lets the OS configure PnP devices not required for boot if your system has a Plug and Play OS.

PCI Latency Timer

This item sets value in units of PCI clocks for PCI device latency timer register. Options are: 32, 64, 96, 128, 160, 192, 224, 248.

Allocate IRQ to PCI VGA
This assigns IRQ to PCI VGA card if card requests IRQ or doesn't assign IRQ to PCI VGA card even if card requests an IRQ.

Palette Snooping
This informs the PCI devices that an ISA graphics device is installed in the system so the card will function correctly.

PCI IDE BusMaster
This uses PCI busmastering for BIOS reading / writing to IDE devices.

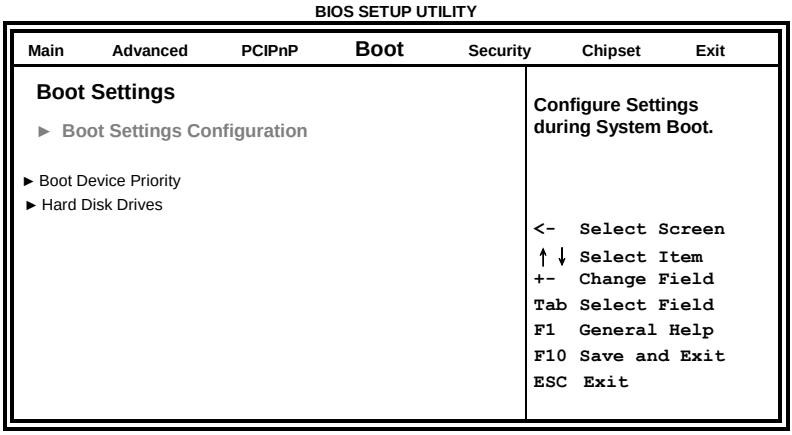
OffBoard PCI/ISA IDE Card
Some PCI IDE cards may require this to be set to the PCI slot number that is holiding the card. ***AUTO***: Works for most PCI IDE cards.

IRQ#
Use the IRQ# address to specify what IRQs can be assigned to a particular peripheral device.

Reserved Memory Size
Size of memory block to reserve for legacy ISA devices.

Boot Settings

This option configures the settings during system boot including boot device priority and HDD/CD/DVD drives.



Boot Settings Configuration

This configuration includes the following items:

Quick Boot - Allows BIOS to skip certain tests while booting. This will decrease the time needed to boot the system.

Quite Boot – ***Disabled***: Displays normal POST messages. ***Enabled***: Displays OEM Logo instead of POST messages.

Bootup Num-Lock – Select Power-on state for Numlock.

PS/2 Mouse Support – Select support for PS/2 Mouse.

Wait for 'F1' If Error – Wait for F1 key to be pressed if error occurs.

Hit 'DEL' Message Display – Displays "Press DEL to run Setup" in POST.

Interrupt 19 Capture – This allows option ROMS to trap interrupt 19.

Boot Device Priority

This specifies the boot sequence from the available devices. A device enclosed in parenthesis has been disabled in the corresponding type menu.

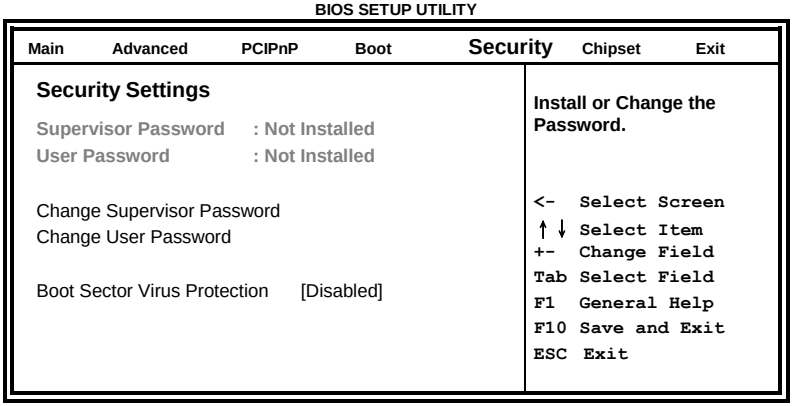
Hard Disk Drives

This specifies the Boot Device Priority sequence from available Hard Drives.

Security Settings

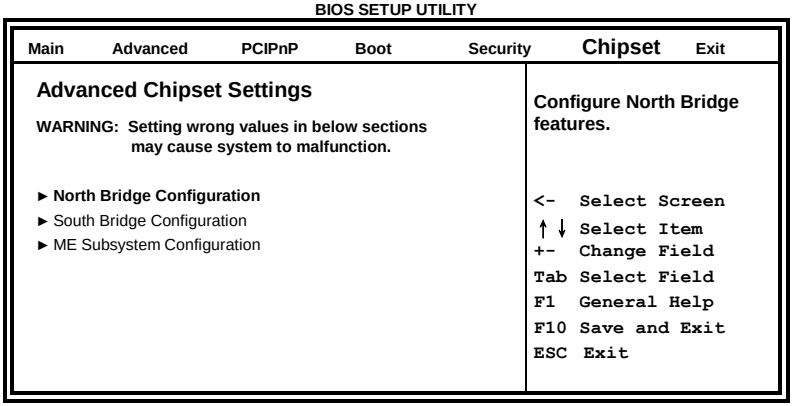
This setting comes with two options set the system password. Supervisor Password sets a password that will be used to protect the system and Setup utility. User Password sets a password that will be used exclusively on the system. To specify a password, highlight the type you want and press <Enter>. The Enter Password: message prompts on the screen. Type the password and press <Enter>. The system confirms your password by asking you to type it again. After setting a password, the screen automatically returns to the main screen.

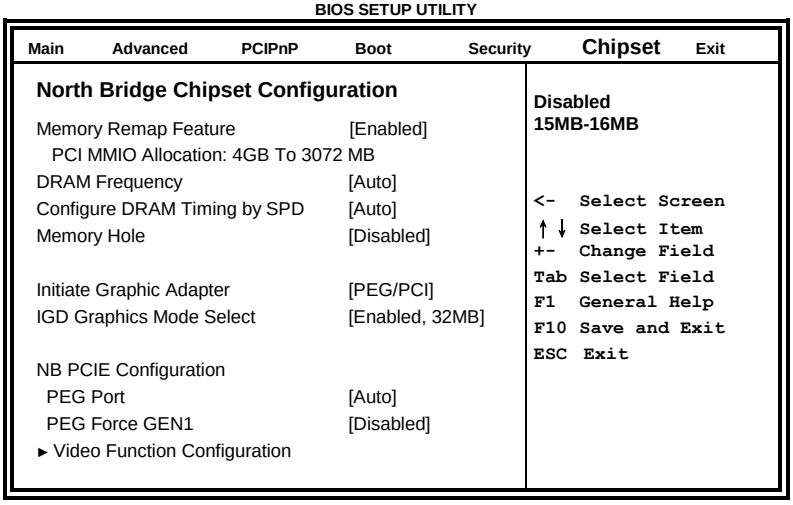
To disable a password, just press the <Enter> key when you are prompted to enter the password. A message will confirm the password to be disabled. Once the password is disabled, the system will boot and you can enter Setup freely.



Advanced Chipset Settings

This setting configures the north bridge, south bridge and the ME subsystem. WARNING! Setting the wrong values may cause the system to malfunction. -


BIOS SETUP UTILITY
Main Advanced PCIPnP Boot Security Chipset Exit
Advanced Chipset Settings
WARNING: Setting wrong values in below sections may cause system to malfunction.
► North Bridge Configuration
► South Bridge Configuration
► ME Subsystem Configuration
Configure North Bridge features.
<- Select Screen
↑↓ Select Item
+- Change Field
Tab Select Field
F1 General Help
F10 Save and Exit
ESC Exit



Memory Remap Feature

This allows remapping of overlapped PCI memory above the total physical memory.

DRAM Frequency

The options are ***Auto***, ***1067 MHz*** and ***1333 MHz***.

Configure DRAM Timing by SPD

The options are ***Auto*** and ***Manual***.

Memory Hole
This option is used to reserve memory space between 15MB and 16MB for ISA expansion cards that require a specified area of memory to work properly.

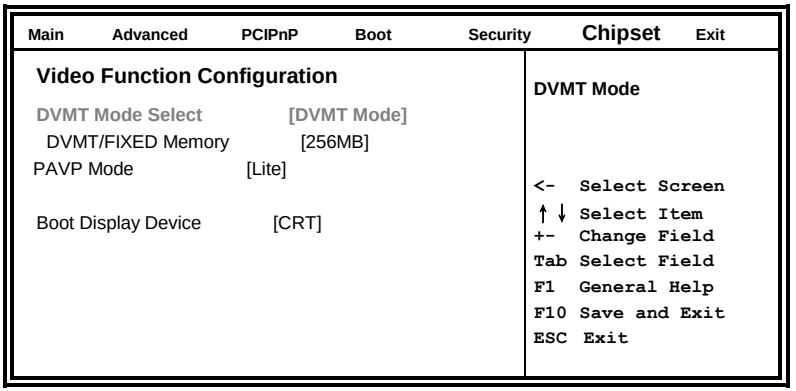
Initiate Graphic Adapter
This option selects which graphics controller to use as the primary boot device.

IGD Graphics Mode Select
This option selects the amount of system memory used by the internal graphics device.

PEG Port
The options are ***Auto*** and ***Disabled***.

PEG Force GEN1
Some non-graphics PCI-E devices may not follow PCI-E specifications and may incorrectly report their GEN capability or link width.

Video Function Configuration
The configuration allows setting to DVMT/FIXED memory.



BIOS SETUP UTILITY

Main	Advanced	PCIPnP	Boot	Security	Chipset	Exit

South Bridge Chipset Configuration

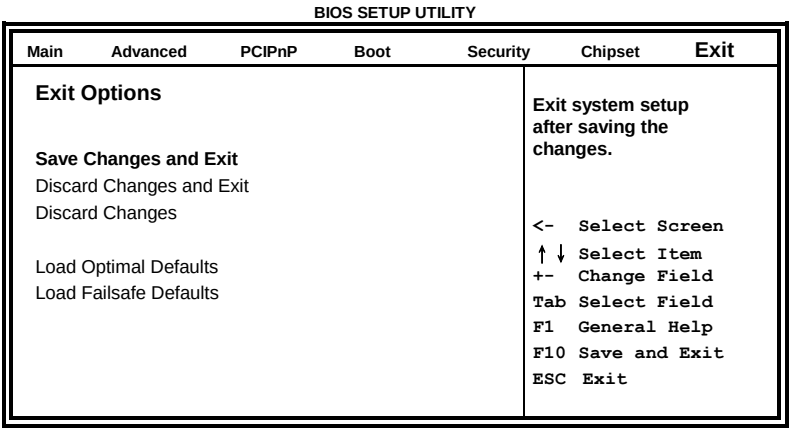
USB Function	[Enabled]
EHCI Controller#1	[Enabled]
EHCI Controller#2	[Enabled]
GbE Controller	[Enabled]
Wake On PCIE LAN	[Enabled]
Wake On RTC Alarm	[Disabled]
SLP_S4# Min. Assertion Width	[4 to 5 seconds]

Enabled
Disabled

<- Select Screen
↑↓ Select Item
+- Change Field
Tab Select Field
F1 General Help
F10 Save and Exit
ESC Exit

Exit Setup

The exit setup has the following settings that are:



Save Changes and Exit

This option allows you to determine whether or not to accept the modifications and save all changes into the CMOS memory before exit.

Discard Changes and Exit

This option allows you to exit the Setup utility without saving the changes you have made in this session.

Discard Changes

This option allows you to discard all the changes that you have made in this session.

Load Optimal Defaults

This option allows you to load the default values to your system configuration. These default settings are optimal and enable all high performance features.

Load Failsafe Defaults

This option allows you to load the troubleshooting default values permanently stored in the BIOS ROM. These default settings are non-optimal and disable all high-performance features.

This page is intentionally left blank.aa

Drivers Installation

This section describes the installation procedures for software and drivers under the Windows. The software and drivers are included with the board. If you find the items missing, please contact the vendor where you made the purchase. The contents of this section include the following:

Intel Chipset Software Installation Utility 53
Intel Graphics Driver Installation 55
LAN Drivers Installation 57
Intel® Management Engine Interface 59

IMPORTANT NOTE:
After installing your Windows operating system, you must install first the Intel Chipset Software Installation Utility before proceeding with the drivers installation.

Intel Chipset Software Installation Utility

The Intel® Chipset Drivers should be installed first before the software drivers to enable Plug & Play INF support for Intel chipset components. Follow the instructions below to complete the. (Before installed Intel Chipset Software Installation Utility.

1. Insert the DVD that comes with the board. Click ***Intel*** and then ***Intel(R) Chipset Software Installation Utility.***

Inside This CD...
Version : 8.5.3D @2
Intel
LAN Card
Tools
Intel(R) Chipset Software Installation Utility
Intel(R) Q57 Chipset Family Graphics Driver
Realtek High Definition Audio Driver
Intel(R) PRO LAN Network Drivers
Intel(R) IAMT 6.0 Drivers
Update Windows OS with Plug and Play feature and allow the OS to

3. When the Welcome screen appears, click ***Next*** to continue.

4. Click ***Yes*** to accept the software license agreement and proceed with the installation process.

5. On the Readme Information screen, click ***Next*** to continue the installation.

6. When the Setup Progress screen appears, click ***Next*** to continue.

Intel® Chipset Device Software
Intel® Chipset Device Software
Setup Progress
Please wait while the following setup operations are performed:
Installing Driver: Intel(R) 5 Series/3400 Series Chipset Family SMBus Controller - 3B30
Version: 1.0.0.0002
Installing Driver: Intel(R) 5 Series/3400 Series Chipset Family USB Enhanced Host Controller -
Version: 9.1.1.1013
Installing Driver: Intel(R) 5 Series/3400 Series Chipset Family USB Enhanced Host Controller -
Version: 9.1.1.1013
Installing Driver: Intel(R) processor DRAM Controller - 0040
Version: 9.1.1.1013
Click Next to continue.
Next
Intel® Installation Framework

7. The Setup process is now complete. Click ***Finish*** then restart the computer and for changes to take effect.

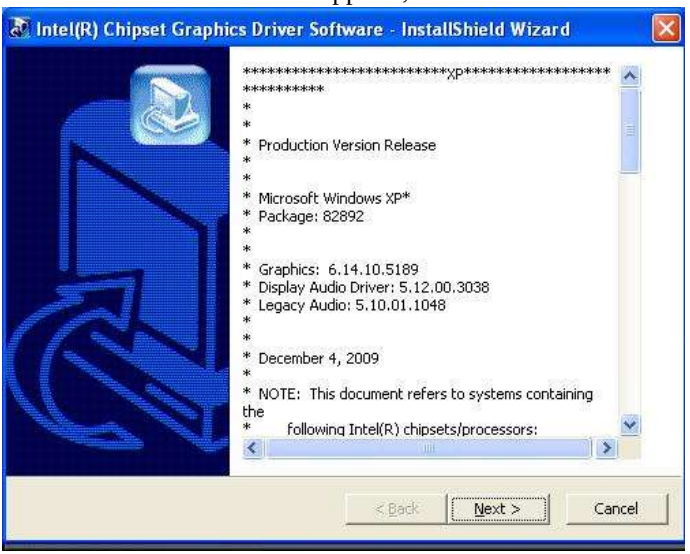
Intel® Chipset Device Software
Intel® Chipset Device Software
Setup Is Complete
You must restart this computer for the changes to take effect. Would you like to restart the computer now?
Yes, I want to restart this computer now.
No, I will restart this computer later.
Click Finish, then remove any installation media from the drives.
Finish
Intel® Installation Framework

Intel Graphics Driver Installation

1. Insert the DVD that comes with the board. Click ***Intel -> Intel® Q57 Chipset Family Graphics Driver***.

2. When the InstallShield Wizard screen appears, click ***Next***.

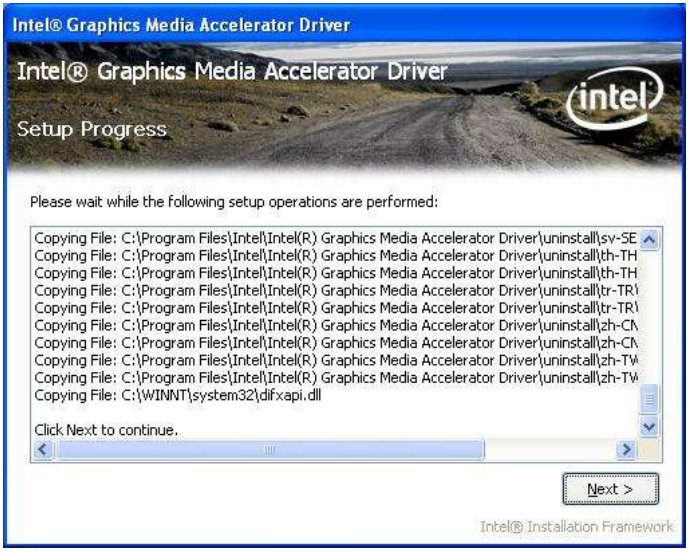
3. When the Welcome screen appears, click ***Next*** to continue.



4. Click ***Yes*** to accept the software license agreement and proceed with the installation process.

5. On Readme File Information screen, click ***Next*** to continue.

6. On Setup Progress screen, click ***Next*** to continue the installation.


Intel® Graphics Media Accelerator Driver
Intel® Graphics Media Accelerator Driver
Setup Progress
intel
Please wait while the following setup operations are performed:
Copying File: C:\Program Files\Intel\Intel(R) Graphics Media Accelerator Driver\uninstall\sv-SE
Copying File: C:\Program Files\Intel\Intel(R) Graphics Media Accelerator Driver\uninstall\th-TH
Copying File: C:\Program Files\Intel\Intel(R) Graphics Media Accelerator Driver\uninstall\th-TH
Copying File: C:\Program Files\Intel\Intel(R) Graphics Media Accelerator Driver\uninstall\tr-TR
Copying File: C:\Program Files\Intel\Intel(R) Graphics Media Accelerator Driver\uninstall\tr-TR
Copying File: C:\Program Files\Intel\Intel(R) Graphics Media Accelerator Driver\uninstall\zh-CN
Copying File: C:\Program Files\Intel\Intel(R) Graphics Media Accelerator Driver\uninstall\zh-CN
Copying File: C:\Program Files\Intel\Intel(R) Graphics Media Accelerator Driver\uninstall\zh-TW
Copying File: C:\Program Files\Intel\Intel(R) Graphics Media Accelerator Driver\uninstall\zh-TW
Copying File: C:\WINNT\system32\difxapi.dll
Click Next to continue.
Next >
Intel® Installation Framework

7. The Setup process is now complete. Click ***Finish*** to restart the computer and for changes to take effect.

LAN Drivers Installation

Follow the steps below to start installing the Intel LAN drivers.

1. Insert the DVD that comes with the board. Click ***Intel*** and then ***Intel(R) PRO LAN Network Drivers***.

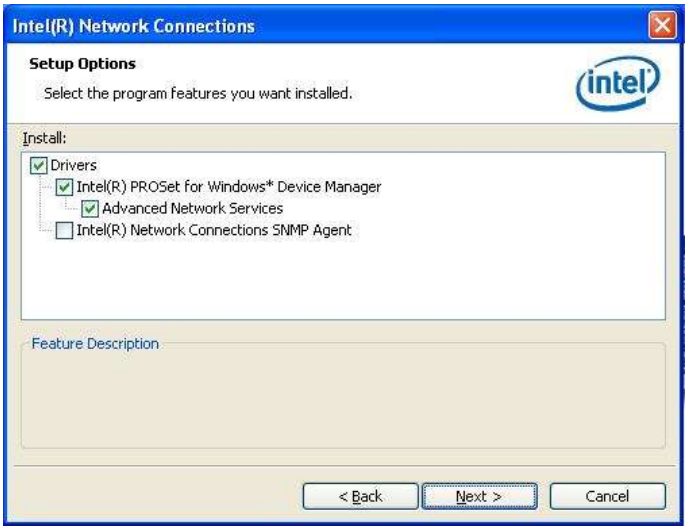
2. Click ***Intel(R) PRO LAN Network Drivers.***

3. On the next screen, click ***Install Drivers*** to start the drivers installation.

4. When the Welcome screen appears, click ***Next*** to continue.

5. In the License Agreement screen, click ***I accept the terms in license agreement*** and ***Next*** to accept the software license agreement and proceed with the installation process.

6. When the Setup Options appears, click ***Drivers*** as shown below and ***Next*** to continue.



7. When the Ready to Install the Program screen appears, click ***Install*** to continue.

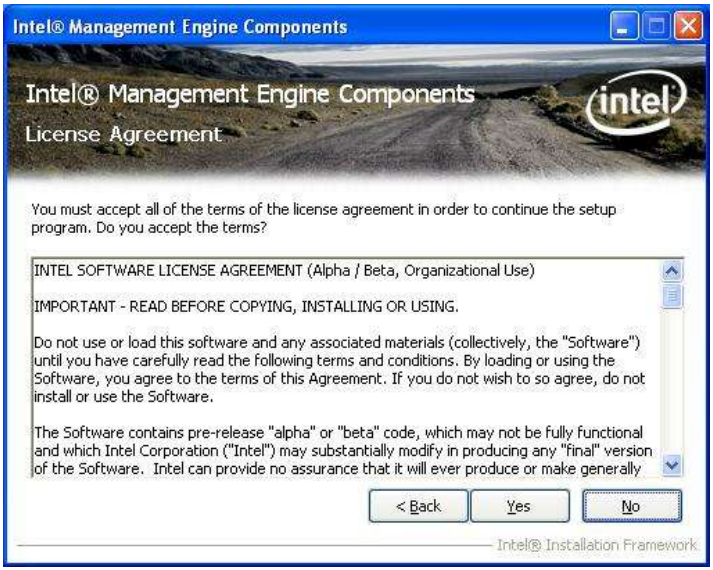
Intel(R) Network Connections - InstallShield Wizard
Ready to Install the Program
The wizard is ready to begin installation.
Click Install to begin the installation.
If you want to review or change any of your installation settings, click Back. Click Cancel to exit the wizard.
InstallShield
< Back
Install
Cancel

8. The Setup process is now complete (InstallShield Wizard Completed). Click ***Finish*** to restart the computer and for changes to take effect.

Intel® Management Engine Interface

1. Insert the drivers disc that comes with the motherboard. Click ***Intel*** and then ***Intel(R) AMT 6.0 Drivers***. When the welcome screen of the Intel® Management Engine Components appears, click ***Next*** to continue. On the next screen, click ***Next*** to agree to the license agreement.

Intel® Management Engine Components
Intel® Management Engine Components
intel
Welcome to the Setup Program
This setup program will install the following components:
- Intel® Management Engine Interface
- Serial Over LAN
It is strongly recommended that you exit all programs before continuing. Click Next to continue.
< Back
Next >
Cancel
Intel® Installation Framework


Intel® Management Engine Components
Intel® Management Engine Components
License Agreement
You must accept all of the terms of the license agreement in order to continue the setup program. Do you accept the terms?
INTEL SOFTWARE LICENSE AGREEMENT (Alpha / Beta, Organizational Use)
IMPORTANT - READ BEFORE COPYING, INSTALLING OR USING.
Do not use or load this software and any associated materials (collectively, the "Software") until you have carefully read the following terms and conditions. By loading or using the Software, you agree to the terms of this Agreement. If you do not wish to so agree, do not install or use the Software.
The Software contains pre-release "alpha" or "beta" code, which may not be fully functional and which Intel Corporation ("Intel") may substantially modify in producing any "final" version of the Software. Intel can provide no assurance that it will ever produce or make generally
< Back
Yes
No
Intel® Installation Framework

2. On the next screen, the Readme File Information shows the system requirements and installation information, click ***Next***.

Intel® Management Engine Components
Intel® Management Engine Components
Readme File Information
Refer to the Readme file below to view the system requirements and installation information.
****
* Copyright (c) 2005-2009 Intel Corporation.
* Intel(R) Active Management Technology
* Intel(R) Management Engine Components
* Local Manageability Service
* Serial Over LAN
****
Local Manageability Service (LMS):
This service enables local applications
running on Intel(R) Active Management
Technology (Intel(R) AMT) device
< Back
Next >
Cancel
Intel® Installation Framework

3. When the Setup Progress screen appears, click ***Next*** to continue. Then, click ***Finish*** when the setup progress has been successfully installed to restart the computer.

Intel® Management Engine Components
Intel® Management Engine Components
Setup Progress
Please wait while the following setup operations are performed:
Copying File: C:\Program Files\Intel\Intel(R) Management Engine Components\uninstall\zh-CN
Copying File: C:\Program Files\Intel\Intel(R) Management Engine Components\uninstall\zh-TW
Copying File: C:\Program Files\Intel\Intel(R) Management Engine Components\uninstall\zh-TW
Copying File: C:\WINNT\system32\difxapi.dll
Installing Driver: Intel(R) Management Engine Interface
Version: 6.0.0.1137
Creating Registry Key: HKLM\SOFTWARE\Intel\AMT\MEIVersion=6.0.0.1137,sz
Installing Driver: Intel(R) Active Management Technology - SOL
Version: 6.0.0.1137
Creating Registry Key: HKLM\SOFTWARE\Intel\AMT\SOLVersion=6.0.0.1137,sz
Click Next to continue.
Next >
Intel® Installation Framework

Intel® Management Engine Components
Intel® Management Engine Components
Setup Is Complete
You must restart this computer for the changes to take effect. Would you like to restart the computer now?
Yes, I want to restart this computer now.
No, I will restart this computer later.
Click Finish, then remove any installation media from the drives.
Finish
Intel® Installation Framework